

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:

PATENT
Attorney Docket No.: 16869S-091000US
Client Ref. No.: W1139-01EY

On 7-22-05

TOWNSEND and TOWNSEND and CREW LLP

By: Elizabeth Neslitt

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Nobuyuki Saika et al.

Application No.: 10/649,124

Filed: August 26, 2003

For: Method and Program for Storing Performance Data, and System Using the Same

Customer No.: 20350

Confirmation No. 4758

Examiner: Safet Metjahic

Technology Center/Art Unit: 2161

RENEWED

PETITION TO MAKE SPECIAL FOR
NEW APPLICATION PURSUANT TO
37 C.F.R. § 1.102(d) &
M.P.E.P. § 708.02, Item VIII,
ACCELERATED EXAMINATION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

A petition to make special the above-identified application in accordance with MPEP § 708.02, Item VIII, accelerated examination was filed April 14, 2005. A decision mailed June 5, 2005 denied said petition.

Submitted herewith is a renewed petition to make special the above-identified application in accordance with MPEP § 708.02, Item VIII, accelerated examination. The application has not received any examination by the Examiner.

(A) The Commissioner is authorized to charge the petition fee of \$130 under 37 C.F.R. § 1.17(h), and any additional fees that may be associated with this petition may be charged to Deposit Account No. 20-1430.

07/25/2005 SDENBOB1 00000057 201430 10649124

01 FC:1464

130.00 DA

(B) All the claims are believed to be directed to a single invention. If the examiner determines that all the claims presented are not obviously directed to a single invention, then Applicant will make an election without traverse as a prerequisite to the grant of special status where the specific grouping of claims will be determined by the examiner.

(C) A pre-examination search was performed by an independent patent search firm on or around March 4, 2005. The classification search covered Class **707**, subclasses 2, 104.1, and 205; Class **709**, subclass 226; and Class **711**, subclass 170. Additionally, a computer database search was conducted on the USPTO full-text database including published U.S. patent applications. The following references were identified in the search report:

(1) U.S. Patent Nos.:

5,796,633	Burgess et al.
6,493,837	Pang et al.

(2) U.S. Patent Application Publication Nos.:

2004/0111725	Srinivasan et al.
2004/0122799	Goyal et al.
2004/0148485	Suzuki

(D) The above references are enclosed herewith, collectively as Exhibit B.

(E) Set forth below is a detailed discussion of the references, pointing out with particularity how the claimed subject matter recited in the claims, amended according to the preliminary amendment filed herewith, is distinguishable over the references.

Claimed Subject Matter of the Present Invention

The pending claims are claims 3-10. Claims 3-7, 9, and 10 are independent claims.

Claim 3 recites a data management method for managing performance data of a computer system. A controller detects free space of a storage area in the computer system. The controller determines a method for storing the performance data depending on the detected free space. The controller acquires performance data. The controller stores the acquired performance data if the free space in the storage area exceeds a preset capacity, and deletes already stored

performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data.

Claim 4 recites a data management method for managing performance data of a computer system. A controller detects free space of a storage area in the computer system. The controller determines a method for storing the performance data depending on the detected free space. The controller acquires performance data. The controller computes the capacity needed to store the acquired performance data. If the free space in the storage area is less than a preset capacity, then an amount of performance data that is already stored on the storage area is deleted; the amount that is deleted is at least the needed capacity. The acquired performance data is then stored.

Claim 5 recites a data management method for managing performance data of a computer system. A controller detects free space of a storage area in the computer system. If the free space exceeds a first preset capacity, then the controller determines a method for storing the performance data depending on the detected free space, acquires performance data, and stores the acquired performance data. If the free space is less than the first preset capacity and the amount of already stored performance data is less than a second preset capacity, then the controller acquires performance data, computes the capacity needed to store the acquired performance data, and if the sum of the needed capacity and the amount of existing data is less than the second present, then the acquired performance data is stored, otherwise enough existing data is deleted so that the sum is less than or equal to the second preset capacity. If the free space is less than the first preset capacity and the existing data exceeds the second preset capacity, then the controller acquires performance data, computes the needed capacity, and deletes enough existing data so that the sum is less than or equal to the second preset capacity.

Claim 6 recites a data management method for managing performance data of a computer system. A controller assigns a preset storage capacity in a storage area for storing performance data. The controller acquires performance data and computes the needed capacity to store the acquired performance data. The controller stores the acquired performance data if the free space in the storage area exceeds the needed capacity. The controller deletes already

stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data.

Claim 7 recites a data management method for managing performance data of a computer system. A controller assigns a preset storage capacity in a storage area for storing performance data. The controller acquires performance data and computes the needed capacity to store the acquired performance data. The controller stores the acquired performance data if the free space in the storage area exceeds the needed capacity. The controller deletes already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data. The amount of already stored performance data that is deleted is enough so that the free space is equal to or greater than the needed capacity.

Claim 9 is recites a controller corresponding to the method of claim 6.

Claim 10 is recites a program for operating a controller corresponding to the method of claim 6.

U.S. Patent No. 5,796,633 Burgess et al.

The patent to Burgess et al. provides for a Method and System for Performance Monitoring in Computer Networks. Discussed is logging thread 50 which logs performance data at a predetermined time interval. Logging thread 50 stores the relevant performance data with a time stamp identification in performance data file 56. Performance data file 56 is a data file that may be stored on disk drives 36 of monitored computer 12. Logging thread 50 monitors performance data relating to physical disks of monitored computer 12, such as the percentage of free space on each logical disk.

The reference does not appear to show or teach the idea of determining if there is sufficient free space before storing performance data, which is a common aspect of each of the independent claims. See for example their discussion of logging thread 50, at column 13, line 61 - column 14, line 19.

With respect to **claim 3**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of

the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data.

With respect to **claim 4**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data, where the amount of data deleted is at least the needed capacity.

With respect to **claim 5**, the reference does not appear to teach or suggest detecting free space of a storage area in the computer system for storing the performance data. The reference does not appear to teach or suggest a “first preset capacity” or a “second preset capacity”. The reference does not appear to teach or suggest determining a method for storing the performance data depending on the detected free space, acquiring performance data, and storing the acquired performance data in the case where the free space exceeds a first preset capacity. The reference does not appear to teach or suggest, in the case where the free space is less than the first preset capacity and the amount of already stored performance data is less than a second preset capacity, acquiring performance data, computing the capacity needed to store the acquired performance data, and if the sum of the needed capacity and the amount of existing data is less than the second present, then storing the acquired performance data, otherwise deleting enough existing data so that the sum is less than or equal to the second preset capacity. The reference does not appear to teach or suggest acquiring performance data, computing the needed capacity, and deleting enough existing data so that the sum is less than or equal to the second preset capacity in the case where the free space is less than the first preset capacity and the existing data exceeds the second preset capacity.

With respect to **claim 6**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest computing the needed capacity to store acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data. See also **claim 9** and **claim 10**.

With respect to **claim 7**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest acquiring performance data and computing the needed capacity to store the acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data, where the amount of already stored performance data that is deleted is enough so that the free space is equal to or greater than the needed capacity.

U.S. Patent No. 6,493,837 Pang et al.

The patent to Pang et al. provides for Using Log Buffers to Trace an Event in a Computer System. Discussed is event tracing program 230 which is stored in memory 201 accessible by one or more processors 200. Event tracing program 230 receives performance data about an event occurring on a computer system from data producer program 226. A user-mode data producer program 226 may be the Microsoft brand Internet Information Server which can detect internet accesses, downloads, etc. To log an event to a log buffer, data producer program 226 first passes the event performance data to event tracing program 230 at step 302. Log buffer 204 may be referenced in list 220 of free log buffers. If log buffer 204 is not full, event tracing program 230 will determine the location at which to start writing the log entry by examining the

current offset value represented by offset variable 206 at step 308. Program 230 will record the performance data into reserved memory block 210 at step 312.

Pang et al. disclose that filled log buffers are placed in a buffer flush list to be written to disk. See for example the Abstract. Fig. 3 also shows a step 318 of adding a buffer to the flush list. Kindly refer to column 5, line 66 to column 6, line 9.

With respect to **claim 3**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data.

With respect to **claim 4**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data, where the amount of data deleted is at least the needed capacity.

With respect to **claim 5**, the reference does not appear to teach or suggest detecting free space of a storage area in the computer system for storing the performance data. The reference does not appear to teach or suggest a “first preset capacity” or a “second preset capacity”. The reference does not appear to teach or suggest determining a method for storing the performance data depending on the detected free space, acquiring performance data, and storing the acquired performance data in the case where the free space exceeds a first preset capacity. The reference does not appear to teach or suggest, in the case where the free space is less than the first preset capacity and the amount of already stored performance data is less than a

second preset capacity, acquiring performance data, computing the capacity needed to store the acquired performance data, and if the sum of the needed capacity and the amount of existing data is less than the second present, then storing the acquired performance data, otherwise deleting enough existing data so that the sum is less than or equal to the second preset capacity. The reference does not appear to teach or suggest acquiring performance data, computing the needed capacity, and deleting enough existing data so that the sum is less than or equal to the second preset capacity in the case where the free space is less than the first preset capacity and the existing data exceeds the second preset capacity.

With respect to **claim 6**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest computing the needed capacity to store acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data. See also **claim 9** and **claim 10**.

With respect to **claim 7**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest acquiring performance data and computing the needed capacity to store the acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data, where the amount of already stored performance data that is deleted is enough so that the free space is equal to or greater than the needed capacity.

U.S. Publication No. 2004/0111725 Srinivasan et al.

The patent application publication to Srinivasan et al. provides for Systems and Methods for Policy-Based Application Management. Discussed is application scheduler 240 receiving at least one policy for managing the applications over computer servers. System 200 includes performance data collector 210. Collector 210 collects usage information indicating performance of the applications, application instances, and/or computer servers 122, 132, 142.

Referring to paragraphs [0040] and [0041], the reference discloses sending such usage information to an application scheduler 150. Policies are applied based on the usage information.

With respect to **claim 3**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data.

With respect to **claim 4**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data, where the amount of data deleted is at least the needed capacity.

With respect to **claim 5**, the reference does not appear to teach or suggest detecting free space of a storage area in the computer system for storing the performance data. The reference does not appear to teach or suggest a “first preset capacity” or a “second preset capacity”. The reference does not appear to teach or suggest determining a method for storing

the performance data depending on the detected free space, acquiring performance data, and storing the acquired performance data in the case where the free space exceeds a first preset capacity. The reference does not appear to teach or suggest, in the case where the free space is less than the first preset capacity and the amount of already stored performance data is less than a second preset capacity, acquiring performance data, computing the capacity needed to store the acquired performance data, and if the sum of the needed capacity and the amount of existing data is less than the second present, then storing the acquired performance data, otherwise deleting enough existing data so that the sum is less than or equal to the second preset capacity. The reference does not appear to teach or suggest acquiring performance data, computing the needed capacity, and deleting enough existing data so that the sum is less than or equal to the second preset capacity in the case where the free space is less than the first preset capacity and the existing data exceeds the second preset capacity.

With respect to **claim 6**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest computing the needed capacity to store acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data. See also **claim 9** and **claim 10**.

With respect to **claim 7**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest acquiring performance data and computing the needed capacity to store the acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data, where the amount of already stored performance data that is deleted is enough so that the free space is equal to or greater than the needed capacity.

U.S. Publication No. 2004/0122799 Goyal et al.

The patent application publication to Goyal et al. provides for a System for Automated Storage Management for Databases. Discussed is policy manager 211 which has a user interface and a policy engine which encodes performance and workload requirements of tablespaces 200 in a policy that is based on user input received through the user interface. Performance information is collected at the disk and storage controllers, transported to policy manager 211 and stored in performance database 202. Databases 202 create tablespaces 200, and databases 202 establish a storage allocation for tablespaces 200 based on the policy. Policy manager 211 detects if tablespace 200 is out of storage and automatically extends the storage allocation for tablespace 200 based on the policy. Storage reallocation processor 415 may reallocate portions of tablespaces 200 and databases 202 to different storage units 310 in order to comply with various rules.

With respect to **claim 3**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data.

With respect to **claim 4**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data, where the amount of data deleted is at least the needed capacity.

With respect to **claim 5**, the reference does not appear to teach or suggest detecting free space of a storage area in the computer system for storing the performance data. The reference does not appear to teach or suggest a “first preset capacity” or a “second preset capacity”. The reference does not appear to teach or suggest determining a method for storing the performance data depending on the detected free space, acquiring performance data, and storing the acquired performance data in the case where the free space exceeds a first preset capacity. The reference does not appear to teach or suggest, in the case where the free space is less than the first preset capacity and the amount of already stored performance data is less than a second preset capacity, acquiring performance data, computing the capacity needed to store the acquired performance data, and if the sum of the needed capacity and the amount of existing data is less than the second present, then storing the acquired performance data, otherwise deleting enough existing data so that the sum is less than or equal to the second preset capacity. The reference does not appear to teach or suggest acquiring performance data, computing the needed capacity, and deleting enough existing data so that the sum is less than or equal to the second preset capacity in the case where the free space is less than the first preset capacity and the existing data exceeds the second preset capacity.

With respect to **claim 6**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest computing the needed capacity to store acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data. See also **claim 9** and **claim 10**.

With respect to **claim 7**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest acquiring performance data and computing the needed capacity to store the acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the

needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data, where the amount of already stored performance data that is deleted is enough so that the free space is equal to or greater than the needed capacity.

U.S. Publication No. 2004/0148485 Suzuki

The patent application publication to Suzuki provides for a System and Method for Managing Storage Device and Program for the Same. Disclosed is storage managing server 101 containing policy definition file 121, scenario definition file 122 and performance information obtaining function 112. Performance information obtaining function 112 is used to obtain information about the utilization rate of the I/O port of the storage devices 231, 232 and fiber channel switches 221, 222, 223, 224 or information about free areas of the disks 231, 232. Scenario parameter definition file 122 describes the conditions to be checked before the execution of scenarios, such as “when the disk free area is under 20%, append disk storage”. Storage management server 101 may make inquiries to the disk devices 231, 232 to collect information on performance and load.

With respect to **claim 3**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data.

With respect to **claim 4**, the reference does not appear to teach or suggest determining a method for storing performance data depending on the detected free space in the storage area. The reference does not appear to teach or suggest the use of a “preset capacity” of the storage area. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds a preset capacity. The reference

does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the preset capacity before storing the acquired performance data, where the amount of data deleted is at least the needed capacity.

With respect to **claim 5**, the reference does not appear to teach or suggest detecting free space of a storage area in the computer system for storing the performance data. The reference does not appear to teach or suggest a “first preset capacity” or a “second preset capacity”. The reference does not appear to teach or suggest determining a method for storing the performance data depending on the detected free space, acquiring performance data, and storing the acquired performance data in the case where the free space exceeds a first preset capacity. The reference does not appear to teach or suggest, in the case where the free space is less than the first preset capacity and the amount of already stored performance data is less than a second preset capacity, acquiring performance data, computing the capacity needed to store the acquired performance data, and if the sum of the needed capacity and the amount of existing data is less than the second present, then storing the acquired performance data, otherwise deleting enough existing data so that the sum is less than or equal to the second preset capacity. The reference does not appear to teach or suggest acquiring performance data, computing the needed capacity, and deleting enough existing data so that the sum is less than or equal to the second preset capacity in the case where the free space is less than the first preset capacity and the existing data exceeds the second preset capacity.

With respect to **claim 6**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference does not appear to teach or suggest computing the needed capacity to store acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data. See also **claim 9** and **claim 10**.

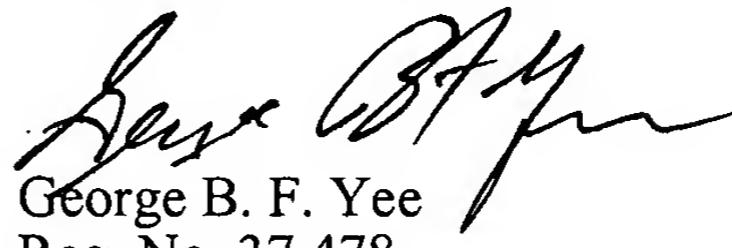
With respect to **claim 7**, the reference does not appear to teach or suggest assigning a preset storage capacity in a storage area for storing performance data. The reference

does not appear to teach or suggest acquiring performance data and computing the needed capacity to store the acquired performance data. The reference does not appear to teach or suggest storing the acquired performance data if the free space in the storage area exceeds the needed capacity. The reference does not appear to teach or suggest deleting already stored performance data if the free space in the storage area does not exceed the needed capacity before storing the acquired performance data, where the amount of already stored performance data that is deleted is enough so that the free space is equal to or greater than the needed capacity.

Conclusion

In view of this comments presented in the instant petition and the claim amendments presented in the accompanying preliminary amendment, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,



George B. F. Yee
Reg. No. 37,478

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: 415-576-0300
Attachments
GBFY
60544009 v1

CONFIDENTIAL
(Patentability Search)

I. SEARCH FEATURE

A. General

Data management

B. Specific

Our search was conducted to find prior art for claims 1-10 of U.S. Application 10/649,124. The claims as generally characterized by a data management method for managing performance data of a computer system which includes: a storage area for storing the performance data as data including performance information of the computer system and other data; and a controller for controlling the storage area, comprising: a step in which the controller detects free space of the storage area; a step in which the controller determines a method for storing the performance data depending on the detected free space; a step in which the controller acquires the performance data; and a step in which the controller stores the acquired performance data in the storage area according to the storing method determined in the method determination step.

C. Application

Performance data management of a computer system

II. FIELD OF SEARCH

The search of the above features was conducted in the following areas:

A. Classification search

<u>Class</u>	<u>Subclasses</u>	<u>Description</u>
707/		DATA PROCESSING: DATABASE AND FILE MANAGEMENT OR DATA STRUCTURES
2		Access augmentation or optimizing
104.1		Application of database or data structure (e.g., distributed, multimedia, image)
205		File allocation

<u>Class</u>	<u>Subclasses</u>	<u>Description</u> (continued)
	709/	ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: MULTICOMPUTER DATA TRANSFERRING
	226	Network resource allocating
	711/	ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: MEMORY
	170	Memory configuring

The above subclasses represent areas deemed to contain subject matter of interest to one or more of the search features. Please note that relevant references may be classified outside of these areas. The integrity of the search is based on the records as presented to us by the United States Patent and Trademark Office (USPTO). No further integrity studies were performed. Also a key word search was performed on the USPTO full-text database including published U.S. patent applications.

III. RESULTS OF SEARCH

A. References developed as a result of search:

<u>U.S. Patent No.</u>	<u>Inventor</u>
5,796,633	Burgess et al.
6,493,837 B1	Pang et al.

<u>U.S. Patent Application Publication No.</u>	<u>Inventor</u>
2004/0111725 A1	Srinivasan et al.
2004/0122799 A1	Goyal et al.
2004/0148485 A1	Suzuki

B. Discussion of related references in numerical order:

The patent to Burgess et al. (5,796,633), assigned to Electronic Data Systems Corporation provides for a *Method and System for Performance Monitoring in Computer Networks*. Discussed is logging thread 50 which logs performance data at a predetermined time interval. Logging thread 50 stores the relevant performance data with a time stamp identification in performance data file 56. Performance data file 56 is a data file that may be stored on disk drives 36 of monitored computer 12. Logging thread 50 monitors performance data relating to physical disks of monitored computer 12, such as the percentage of

free space on each logical disk. The reference appears to fail to either anticipate or render obvious at least the following limitation of applicants' claimed invention: modification of performance data depending on the detected free disk space (see figures 1-3; column 8, lines 20-21, 30-32, 63-67; and column 9, lines 1-3).

The patent to Pang et al. (6,493,837 B1), assigned to Microsoft Corporation provides for *Using Log Buffers to Trace an Event in a Computer System*. Discussed is event tracing program 230 which is stored in memory 201 accessible by one or more processors 200. Event tracing program 230 receives performance data about an event occurring on a computer system from data producer program 226. A user-mode data producer program 226 may be the Microsoft brand Internet Information Server which can detect internet accesses, downloads, etc. To log an event to a log buffer, data producer program 226 first passes the event performance data to event tracing program 230 at step 302. Log buffer 204 may be referenced in list 220 of free log buffers. If log buffer 204 is not full, event tracing program 230 will determine the location at which to start writing the log entry by examining the current offset value represented by offset variable 206 at step 308. Program 230 will record the performance data into reserved memory block 210 at step 312. The reference appears to fail to either anticipate or render obvious at least the following limitation of applicants' claimed invention: modification of performance data depending on the detected free storage space (see figures 2, 3; column 4, lines 48-50, 60-65; column 5, lines 3-5, 7-9, 27-29, 45-47; and column 6, lines 27-30 and 50-52).

The patent application publication to Srinivasan et al. (2004/0111725 A1) provides for *Systems and Methods for Policy-Based Application Management*. Discussed is application scheduler 240 receiving at least one policy for managing the applications over computer servers. System 200 includes performance data collector 210. Collector 210 collects usage information indicating performance of the applications, application instances, and/or computer servers 122, 132, 142. The reference appears to fail to either anticipate or render obvious at least the following limitations of applicants' claimed invention: detection of free storage space and modification of performance data depending on the detected free storage space (see figures 1, 2; abstract; paragraphs 27 and 28).

The patent application publication to Goyal et al. (2004/0122799 A1) provides for a *System for Automated Storage Management for Databases*. Discussed is policy manager 211 which has a user interface and a policy engine which encodes performance and workload requirements of tablespaces 200 in a policy that is based on user input received through the user interface. Performance information is collected at the disk and storage controllers, transported to policy manager 211 and stored in performance database 202. Databases 202 create tablespaces 200, and databases 202 establish a storage allocation for tablespaces 200 based on the policy. Policy manager 211 detects if tablespace 200 is out of storage and automatically extends the storage allocation

for tablespace 200 based on the policy. Storage reallocation processor 415 may reallocate portions of tablespaces 200 and databases 202 to different storage units 310 in order to comply with various rules. The reference appears to fail to either anticipate or render obvious at least the following limitation of applicants' claimed invention: a relationship between storing performance data and detection of free storage space (see figures 2, 3, 4; abstract; and paragraphs 16, 26, 28, 30, 40, and 44).

The patent application publication to Suzuki (2004/0148485 A1) provides for a *System and Method for Managing Storage Device and Program for the Same*. Disclosed is storage managing server 101 containing policy definition file 121, scenario definition file 122 and performance information obtaining function 112. Performance information obtaining function 112 is used to obtain information about the utilization rate of the I/O port of the storage devices 231, 232 and fiber channel switches 221, 222, 223, 224 or information about free areas of the disks 231, 232. Scenario parameter definition file 122 describes the conditions to be checked before the execution of scenarios, such as "when the disk free area is under 20%, append disk storage". Storage management server 101 may make inquiries to the disk devices 231, 232 to collect information on performance and load. The reference appears to fail to either anticipate or render obvious at least the following limitation of applicants' claimed invention: a relationship between storing performance data and detection of free storage space (see figures 1, 2, 8; abstract; and paragraphs 21, 22, 24, 38, and 45).

Sejal Gangar

TABLE OF CONTENTS

U.S. Patent No. 5,796,633 to Burgess et al.	1
U.S. Patent No. 6,493,837 to Pang et al.	2
U.S. Patent Application Publication No. 2004/0111725 to Srinivasan et al.	3
U.S. Patent Application Publication No. 2004/0122799 to Goyal et al.	4
U.S. Patent Application Publication No. 2004/0148485 to Suzuki	5

60466452 v1